

Claims

WHAT IS CLAIMED IS:

1 1. A method of providing remote computing services executed upon a server having
2 mass data storage and network communications to remote clients, the method comprising:
3 storing a source code file within the mass storage of the server;
4 receiving a processing service request from a remote client corresponding to a requested
5 processing service specified within the source code file;
6 activating a data processing object corresponding to the processing service requested from the
7 remote client to generate a data response corresponding to the processing service request;
8 formatting the data response into a data response packet; and
9 transmitting the text passed data response packet to the remote client in response to the
10 processing service request,
11 wherein the data processing object corresponds to a compiled data processing class for the
12 requested processing service contained within the source code file.

2 2. A method according to claim 1, wherein the server receives the processing service
3 request having a packet payload body data block and transmits the text based data response packet
4 using an HTTP communications protocol.

1 3. A method according to claim 2, wherein the source code file is stored within the mass
2 storage of the server at a location corresponding to a URL for the processing service request.

1 4. The method according to claim 2, wherein the activating step further comprises:

2 extracting from the packet payload body data block an ID corresponding to the requested
3 processing service;
4 extracting from the packet payload body data block one or more input arguments to be passed
5 to the data processing object;
6 obtaining the data processing object corresponding to the requested processing service; and
7 causing the data processing object to generate the data response based upon the one or more
8 input arguments.

1 5. The method according to claim 4, wherein the obtaining step further comprises:
2 determining whether the data processing object corresponding to the requested processing
3 service is located within a web services library;
4 determining whether any data processing object found within the web services library
5 corresponds to the data processing object generated by compiling the data processing class for the
6 requested processing service contained within the source code file;
7 if the data processing object does not correspond to the data processing object generated by
8 compiling the data processing class for the requested processing service contained within the source
9 code file, automatically compiling the source code file to generate the data processing object for the
10 requested processing service.

1 6. The method according to claim 5, wherein the obtaining step further comprises
2 storing the data processing object within the web services library following the compilation of the
3 source code file for use by subsequent processing service requests.

1 7. The method according to claim 6, wherein the data processing object generates the
2 data response based upon data retrieved from one or more remote databases specified by the data
3 processing the one or more input arguments.

1 8. The method according to claim 6, wherein the data processing object may activate and
2 call one or more other data processing objects on the server.

1 9. The method according to claim 8, wherein the one or more data processing objects
2 correspond to compiled data processing classes contained within source code files which are not the
3 same as the source code file for the data processing class.

1 10. The method according to claim 9, wherein the one or more data processing objects are
2 dynamically created whenever the one or more data processing objects stored within the web services
3 library do not correspond to a compiled version of the appropriate data processing classes.

1 11. A computer program data product readable by a computing system and encoding
2 instructions for providing remote computing services executed upon a server having mass data
3 storage and network communications to remote clients, the computer process comprising:

4 storing a source code file within the mass storage of the server;

5 receiving a processing service request from a remote client corresponding to a requested
6 processing service specified within the source code file;

7 activating a data processing object corresponding to the processing service requested from the
8 remote client to generate a data response corresponding to the processing service request;

9 formatting the data response into a text based data response packet; and

10 transmitting the text passed data response packet to the remote client in response to the
11 processing service request;

12 wherein the data processing object corresponds to a compiled data processing class for the
13 requested processing service contained within the source code file.

1 12. A computer program data product according to claim 11, wherein the server receives
2 the processing service request having a packet payload body data block and transmits the text based
3 data response packet using an HTTP communications protocol.

1 13. A computer program data product according to claim 12, wherein the source code file
2 is stored within the mass storage of the server at a location corresponding to a URL for the
3 processing service request.

4 14. The computer program data product according to claim 12, wherein the activating
5 step further comprises:

6 extracting from the packet payload body data block an ID corresponding to the requested
7 processing service;

8 extracting from the packet payload body data block one or more input arguments to be passed
9 to the data processing object;

10 obtaining the data processing object corresponding to the requested processing service; and

11 causing the data processing object to generate the data response based upon the one or more
12 input arguments.

1 15. The computer program data product according to claim 14, wherein the obtaining step
2 further comprises:

3 determining whether the data processing object corresponding to the requested processing
4 service is located within a web services library;

5 determining whether any data processing object found within the web services library
6 corresponds to the data processing object generated by compiling the data processing class for the
7 requested processing service contained within the source code file; and

8 if the data processing object does not corresponds to the data processing object generated by
9 compiling the data processing class for the requested processing service contained within the source
10 code file, automatically compiling the source code file to generate the data processing object for the
11 requested processing service.

16. The computer program data product according to claim 15, wherein the obtaining step
further comprises storing the data processing object within the web services library following the
compilation of the source code file for use by subsequent processing service requests.

17. The computer program data product according to claim 16, wherein the data
processing object generates the data response based upon data retrieved from one or more remote
databases specified by the data processing the one or more input arguments.

18. The computer program data product according to claim 16, wherein the data
processing object may activate and call one or more other data processing objects on the server.

19. The computer program data product according to claim 18, wherein the one or more
data processing objects are correspond to compiled data processing classes contained within source
code files which are not the same as the source code file for the data processing class.

1 20. The computer program data product according to claim 19, wherein the one or more
2 data processing objects are dynamically created whenever the one or more data processing objects
3 stored within the web services library do not correspond to a compiled version of the appropriate
4 data processing classes.

1 21. A network server for providing remote computing services executed upon the network
2 server having mass data storage for storing a source code file containing a data processing class
3 corresponding to a data processing object that implements a requested remote processing service, the
4 network server comprising:

5 a network interface I/O module for receiving a processing service request from a remote
6 client corresponding to a requested processing service specified within a source code file and for
7 transmitting the text passed data response packet to the remote client in response to the processing
8 service request; and

9 a request processing module for generating a data response corresponding to the processing
10 service request, the request processing module activates the data processing object corresponding to
11 the requested processing service and formatting the data response into a text based data response
12 packet;

13 wherein the data processing object corresponds to a compiled data processing class for the
14 requested processing service contained within the source code file.

1 22. A network server according to claim 21, wherein the server receives the processing
2 service request having a packet payload body data block and transmits the text based data response
3 packet using an HTTP communications protocol.

1 23. A network server according to claim 22, wherein the source code file is stored within
2 the mass storage of the server at a location corresponding to a URL for the processing service
3 request.

1 24. The network server according to claim 22, wherein the request processing module
2 further comprises:

3 extracts from the packet payload body data block an ID corresponding to the requested
4 processing service;

5 extracts from the packet payload body data block one or more input arguments to be passed
6 to the data processing object;

7 obtains the data processing object corresponding to the requested processing service; and
8 causes the data processing object to generate the data response based upon the one or more
9 input arguments.

1 25. The network server according to claim 24, wherein the request processing module
2 obtains the data processing object from a class compiler module, the class compiler module
3 comprises:

4 an obtain/parse web method module for determining whether the data processing object
5 corresponding to the requested processing service is located within a web services library;

6 a compile class module for compiling the source code file to generate the data processing
7 object; and

8 a web method attribute module for identifying one or more processing services made
9 available to remote clients by data processing objects generated when a source code file is compiled;

10 wherein the obtain/parse web method module also determines whether any data processing
11 object found within the web services library corresponds to the data processing object generated by
12 compiling the data processing class for the requested processing service contained within the source
13 code file; and

14 if the data processing object does not corresponds to the data processing object generated by
15 compiling the data processing class for the requested processing service contained within the source
16 code file, the compile class module automatically compiles the source code file to generate the data
17 processing object for the requested processing service.

1 26. The network server according to claim 25, wherein the compiler module further
comprises a cache compiled object module for storing the data processing object within the web
services library following the compilation of the source code file for use by subsequent processing
service requests.

2 27. The network server according to claim 26, wherein the data processing object
generates the data response based upon data retrieved from one or more remote databases specified
by the data processing the one or more input arguments.

1 28. The method according to claim 26, wherein the data processing object may activate
2 and call one or more other data processing objects on the server.

1 29. The method according to claim 28, wherein the one or more data processing objects
2 are correspond to compiled data processing classes contained within source code files which are not
3 the same as the source code file for the data processing class.

1 30. The method according to claim 29, wherein the one or more data processing objects
2 are dynamically created whenever the one or more data processing objects stored within the web
3 services library do not correspond to a compiled version of the appropriate data processing classes.

1 31. A method for automatically compiling server executed source code corresponding to
2 data processing objects used to provide remote processing services upon receipt of a request for a
3 remote processing services, the method comprising:

4 storing a source code file within the mass storage of the server;

5 receiving a processing service request from a remote client corresponding to a requested
6 processing service specified within the source code file;

7 extracting from the packet payload body data block an ID corresponding to the requested
8 processing service;

9 determining whether the data processing object corresponding to the requested processing
10 service is located within a web services library;

11 determining whether any data processing object found within the web services library
12 corresponds to the data processing object generated by compiling the data processing class for the
13 requested processing service contained within the source code file; and

14 if the data processing object does not corresponds to the data processing object generated by
15 compiling the data processing class for the requested processing service contained within the source
16 code file, automatically compiling the source code file to generate the data processing object for the
17 requested processing service.

1 32. The method according to claim 31, wherein the method further comprises storing the
2 data processing object within the web services library following the compilation of the source code
3 file for use by subsequent processing service requests.

1 33. The method according to claim 34, wherein the data processing object may activate
2 and call one or more other data processing objects on the server.

1 34. The method according to claim 33, wherein the one or more data processing objects
2 are correspond to compiled data processing classes contained within source code files which are not
3 the same as the source code file for the data processing class.

 35. The method according to claim 34, wherein the one or more data processing objects
are dynamically created whenever the one or more data processing objects stored within the web
services library do not correspond to a compiled version of the appropriate data processing classes.

 36. A computer program data product readable by a computing system and encoding
instructions for automatically compiling server executed source code corresponding to data
processing objects used to provide remote processing services upon receipt of a request for a remote
processing services, the computer process comprising:

 storing a source code file within the mass storage of the server;

 receiving a processing service request from a remote client corresponding to a requested
processing service specified within the source code file;

 extracting from the packet payload body data block an ID corresponding to the requested
processing service;

10 determining whether the data processing object corresponding to the requested processing
11 service is located within a web services library;

12 determining whether any data processing object found within the web services library
13 corresponds to the data processing object generated by compiling the data processing class for the
14 requested processing service contained within the source code file; and

15 if the data processing object does not corresponds to the data processing object generated by
16 compiling the data processing class for the requested processing service contained within the source
17 code file, automatically compiling the source code file to generate the data processing object for the
18 requested processing service.

37. The computer program data product according to claim 36, wherein the method
further comprises storing the data processing object within the web services library following the
compilation of the source code file for use by subsequent processing service requests.

38. The computer program data product according to claim 37, wherein the data
processing object may activate and call one or more other data processing objects on the server.

39. The computer program data product according to claim 38, wherein the one or more
data processing objects are correspond to compiled data processing classes contained within source
code files which are not the same as the source code file for the data processing class.

40. The computer program data product according to claim 39, wherein the one or more
data processing objects are dynamically created whenever the one or more data processing objects
stored within the web services library do not correspond to a compiled version of the appropriate
data processing classes.

1 41. A method for automatically creating data exchange schema data on a network server
2 corresponding to remote processing services provided by the network server for source code
3 corresponding to data processing objects used to provide the remote processing services upon receipt
4 of a request from a client process, the method comprising:

5 storing a source code file within the mass storage of the server;
6 compiling the source code file to generate a data processing object; and
7 automatically generating the data exchange schema data for the data processing object
8 generated when the source code file is compiled to generate the data processing object that provides
9 the requested processing service.

10 42. The method according to claim 41, wherein the method further comprises storing the
11 data exchange schema data within the web services library for use by subsequent processing service
12 requests.

13 43. The method according to claim 42, wherein data exchange schema data comprises an
14 HTML representation for a web page containing a description of exposed data processing services.

15 44. The method according to claim 43, wherein the web page comprises:
16 a textual description of each exposed data processing service based upon data stored within
17 the source code file;
18 a description of each input argument accepted by each exposed data processing service, the
19 description includes a description of the input argument and a description of the data format for the
20 input argument data expected by the exposed data processing service; and
21 a description of each output data value generated by each exposed data processing service.

1 45. The method according to claim 44, wherein the description of each input argument
2 further comprises an input field upon the generated web page for permitting a user to input a value to
3 be passed to the exposed data processing service as the corresponding input argument.

1 46. The method according to claim 45, wherein the description of each output data value
2 generated by each exposed data processing service further comprises an activate button which causes
3 the remote data processing service to be activated using the values contained within the input fields
4 corresponding to the input arguments as the input arguments submitted with the remote data
5 processing service request.

1 47. The method according to claim 42, data exchange schema data comprises a
2 specification for the input and output data schema expressed in a data transfer specification language.

1 48. The method according to claim 47, wherein the data transfer specification language
2 comprises a Web Services Description Language representation for the data exchange schema data.

1 49. The method according to claim 47, wherein the data transfer specification language
2 comprises a Resource Description Format representation for the data exchange schema data.

1 50. A computer program data product readable by a computing system and encoding
2 instructions for automatically creating data exchange schema data on a network server corresponding
3 to remote processing services provided by the network server for source code corresponding to data
4 processing objects used to provide the remote processing services upon receipt of a request from a
5 client process, the computer process comprising:

6 storing a source code file within the mass storage of the server;

7 compiling the source code file to generate a data processing object; and
8 automatically generating the data exchange schema data for the data processing object
9 generated when the source code file is compiled to generate the data processing object that provides
10 the requested processing service.

1 51. The computer program data product according to claim 50, wherein the method
2 further comprises storing the data exchange schema data within the web services library for use by
3 subsequent processing service requests.

1 52. The computer program data product according to claim 51, wherein data exchange
2 schema data comprises an HTML representation for a web page containing a description of exposed
3 data processing services.

1 53. The computer program data product according to claim 52, wherein the web page
2 comprises:

3 a textual description of each exposed data processing service based upon data stored within
4 the source code file;

5 a description of each input argument accepted by each exposed data processing service, the
6 description includes a description of the input argument and a description of the data format for the
7 input argument data expected by the exposed data processing service; and

8 a description of each output data value generated by each exposed data processing service.

1 54. The computer program data product according to claim 53, wherein the description of
2 each input argument further comprises an input field upon the generated web page for permitting a

3 user to input a value to be passed to the exposed data processing service as the corresponding input
4 argument.

1 55. The computer program data product according to claim 54, wherein the description of
2 each output data value generated by each exposed data processing service further comprises an
3 activate button which causes the remote data processing service to be activated using the values
4 contained within the input fields corresponding to the input arguments as the input arguments
5 submitted with the remote data processing service request.

1 56. The computer program data product according to claim 51, data exchange schema
2 data comprises a specification for the input and output data schema expressed in a data transfer
3 specification language.

4 57. The method according to claim 56, wherein the data transfer specification language
5 comprises a Web Services Description Language representation for the data exchange schema data.

6 58. The method according to claim 56, wherein the data transfer specification language
7 comprises a Resource Description Format representation for the data exchange schema data.